ስለዚህ መጽሐፍ

ፍቃድ

የመጨረሻ ስሪት

የመጽሐፉን የቅርብ ጊዜ ምንጭ ከዚህ ማግኘት ይቻላል ፤
https://github.com/karlseguin/the-little-go-book

መግቢያ

ሁሌም አዲስ የፕሮግራሚንግ ቋንቋ ለመማር ስነሳ ድብልቅልቅ ስሜት ነው የሚሰማኝ ። በአንድ በኩል ፤ ቋንቋ (ፕሮግራሚንግ) ለምንሰራው ነገር በጣም መሰረታዊ ነው ፤በጣም አነስተኛዋ ለውጥ በጣም ከፍተኛ የሚባል ተጽእኖ ልትፈጥር ትችላለች። ፕሮግራም ስንጽፍ የሚገጥሙን አስተማሪ አጋጣሚዎች ወደፊት ለሚኖረን የፕሮግራሚንግ ስልት ዘላቂ የሆነ ተጽእኖ ሊፈጥሩብን እንዲሁም ከሌሎች ቋንቋዎችም የምንጠብቃቸውን ነግሮች እንድናገናዝብ ያደርጉናል። በሌላ በኩል ደግሞ የቋንቋዎች ዲዛይን በአብዛኛው ጭማሪ ጽንሰ-ሃሳቦችን ይዘው ነው የሚመጡት ። ስለዚህ አዲስ ቋንቋ ለመማር ፤ አዳዲስ ቁልፍ ቃላትን፣ የአይነት ስርዓት ፣ የኮድ ስልት ፣ ላይብረሪዎች ፣ ኮሚኒቲዎች እንዲሁም አዲስ አስተሳሰብ መለማመድ ከባድ ስራ ሊመስል ይችላል። ከሌሎች ትምህርቶች አንጻርም አዲስ ቋንቋ መማር ላይ የምናጠፋው ጊዘ የባከነ መስሎ ሊሰማን ይችላል።

ይህን ካልኩ በኋላ ወደፊት ስንቀጥል ፤ አዲስ ቋንቋ በመማር ደረጃ በደረጃ ጭማሪ ለውጦችን ለመውሰድ ፍቃደኛ መሆን አለብን ፤ ምክንያቱም የፕሮግራሚንም ቋንቋዎች ለምንሰራቸው ነገሮች መሰረት ናቸው ። ለውጦቹ ቅጥልጣይ ጭማሪዎች ቢሆኑም ፤ ሰፊ አድማስ አላቸው ። በዚህም ምክንያት ምርታማነት (productivity) ፣ ተነባቢነት (readability) ፣ ውጠታማነት (performance) ፣ ተሞካሪነት (testability) ፣ የኮድ ጥገኝነት (dependency management) ፣ ጥርሰት አያያዝ (error handling) ፣ አሰናነድ (documentation) ፣ ትንታነ (profiling) ፣ ማህበረሰብ (communities) ፣ መነሻ ላይብረሪ (standard libraries) እና የመሳሰሉት ላይ ተጽእኖ ያሳርፋሉ። ምናልባትም እንደዚህ አይነት ለውጦችን እንደመራራ መድሐኒት ልንወስድ እንችላለን?

ይህ ወደዋናው ጥያቄ ያመራናል ፤ ጎ ለምን አስፈለገ ? በእነ እይታ ሁለት አንገብጋቢ ምክንያቶች አሉ ። የመጀመሪያው ምክንያት ፤ በአንፃሩ ቀላል ቋንቋ ስለሆነና መነሻ ላይብረሪውም ቀላል ስለሆነ ነው ። በብዙ መልኩ ሲታይ ፤ ጎ ላይ ያሉት ለውጦች ባለፉት ሁለት አስርት አመታት በፕሮግራሚንግ ቋንቋዎች ላይ እየጨመረ የነበረውን ውስብስብነት የሚያቃልሉ ናቸው ። ለላው ምክንያት ደግሞ ለብዙ ገንቢዎች (developers) ጎ ተጨማሪ አቅም ስለሚሰጥ ነው ።

ጎ የተገነባው እንደ ሲስተም ቋንቋ (ምሳለ ፦ ኦፐሬቲንግ ሲስተም ፣ ዲቫይስ ድራይቨር) የ C እና የ C++ ገንቢዎችን ታሳቢ አድርጎ ነው ። ነገር ግን ጎ-ቲም (Go team) ባለው መረጃ መሰረት የ ጎ ዋና ተጠቃሚዎች የሆኑት የሲስተም ገንቢዎች ሳይሆኑ የ ትግበራ(application) ገንቢዎች ናቸው ። ለምን ? የሲስተም ገንቢዎችን ወክየ መናገር አልችልም ፤ ነገር ግን እንደነ ድረ-ገጽ ለሚገነቡ ፣ ለግልጋሎት (services) ፣ ለደስክቶፕ ትግበራዎች እና ለመሳሰሉት እየተፈጠረ ያለውን በከፍተኛ-እርከንና (higher-level) በዝቅተኛ እርከን (low-level) የሲስተም ትግበራዎች መካከል ያሉ ሲስተሞች ተፈላጊነት እያሚላ ስለሆነ ይመስለኛል ።

ምናልባትም ለመላላኪያ (messaging) ፣ ማቆያ (caching) ፣ ከፍተኛ ስለት ለሚያስፈልገው ውህብ ትንታነ ፣ ለትእዛዝ መስመር መግቢያ (command line interface) ፣ መመዝገቢያ (logging) ወይም መቆጣጠሪያ (monitoring) የተመቸ ቋንቋ ነው ልንል እንችላለን ። እነ ባለኝ የስራ ልምድ የሲስተሞች ውስብስብነት እየጨመረ ሲሀድና ኮንከረንሲ በሺዎች መቆጠር ሲጀምር ፤ ይህንን ሊፈታ የሚችል የኢንፍራስትራክቸር ሲስተም ያስፈልጋል ። እንደ ሩቢ ወይም ፓይተን ያሉ ቋንቋዎችን በመጠቀም እንደዚህ አይነት ሲስተሞች መገንባት የተለመደ ነው ፤ ግን ለእነዚህ ሲስተሞች የጎ ቋሚ የአይነት ስርዓት (type system) እና የተሻለ ውጠታማነት የበለጠ ያሻሽላቸዋል ። እርግጥ ነው ጎ ን በመጠቀም ድረ-ገጽ መገንባት የተለመደ ነው ፤ እንደነ አስተያየት ግን ለዚህ አላማ ኖድ ወይም ሩቢ የተሻለ ተመራጭ ናቸው ብየ አምናለሁ ።

ጎ የላቀ ጥቅም የሚያስገኝባቸው ለሎች ቦታዎች አሉ ። ለምሳለ ፤ ኮምፓይልድ የሆነ የ ጎ ፕሮግራም ስናስፈጽም ምንም አይነት ቅደመ ሁነታ ወይም ጥገኝነት አይኖርም ።

ለማነጻጸር ያህል ለምሳለ የሩቢ ወይም የጃቫ ፕሮግራሞች የተጫነው የሩቢ ወይም የ JVM ስሪት ላይ ይወሰናሉ ። በዚህ ምክንያት ጎ

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println("Hello World!")
7 }
```